AIP Conference Proceedings

Optimizing the throughput of an M/G/C/C topological network

Ruzelan Khalid, Nurhanis Ishak, Md Azizul Baten, and Mohd. Kamal Mohd. Nawawi

Citation: AIP Conference Proceedings **1691**, 030015 (2015); doi: 10.1063/1.4937034
View online: http://dx.doi.org/10.1063/1.4937034
View Table of Contents: http://scitation.aip.org/content/aip/proceeding/aipcp/1691?ver=pdfcov
Published by the AIP Publishing

Optimizing the Throughput of an *M/G/C/C* Topological Network

Ruzelan Khalid[1, a)], Nurhanis Ishak[2, b)], Md Azizul Baten[3, c)] and Mohd. Kamal Mohd. Nawawi[4, d)]

[1, 3, 4]*Department of Decision Sciences, School of Quantitative Sciences, Universiti Utara Malaysia,Sintok, Kedah, Malaysia*
[2]*Faculty of Management and Information Technology, Kolej Universiti Islam Sultan Azlan Shah, Kuala Kangsar, Perak, Malaysia*

[a)]Corresponding author: ruzelan@uum.edu.my
[b)]nurhanis@kuisas.edu.my
[c)]md.azizul@uum.edu.my
[d)]mdkamal@uum.edu.my

Abstract. An *M/G/C/C* state dependent queuing network approach models the behavior of entities (e.g., pedestrians and vehicles) flowing through a space. For a space consisting of various complex topological networks, the challenge is how to control such flow so that its throughput can be optimized. This paper discusses the throughput optimization of an *M/G/C/C* system using a network flow programming model. For this, we first decompose each available network and analyze its performance separately. The performance reports the effect of arrival rates to the throughput, blocking probability, expected number of pedestrians and expected travel time of each of the network. The best arrival rate of each network is then searched and fed to the network flow programming model to find the optimal arrival rates of source networks. The effect of the optimal arrival rates to the performance of the whole network is then analyzed and compared to that of using arbitrary arrival rates. Results show that with the right control of arrival rates, pedestrians can smoothly and efficiently be flowed through a network.

INTRODUCTION

An *M/G/C/C* state dependent queuing network [1-6] has long been used to model the behavior of entities (e.g., pedestrians and vehicles) flowing through a network. Entities are modeled so that their flow speeds are dynamically adjusted according to its current number of residing entities which is associated with the arrival rate of entities to the network. Thus, the arrival rate principally controls various performance measures such the throughput, blocking probability, expected number of entities and expected travel time of the network. Its maximum throughput can also be analyzed by computing its optimal arrival rate. However, the challenge is how to obtain the optimal arrival rates of entities arriving at various arrival sources and flowing through a topological network which eventually maximizes its throughput.

This paper extends our previous paper [7] and discusses how to find the optimal arrival rates to a topological network consisting of series, splitting and merging networks. For this, we first decompose each network and separately analyze the effect of relevant arrival rates to its performance measures. We next search the best arrival rate of each of the available network and fed them to the network flow programming model. By solving this model, we will obtain the optimal arrival rates to source networks maximizing the throughput of the whole network. The effect of the optimal arrival rates to its throughput, blocking probability, expected number of occupants and

Innovation and Analytics Conference and Exhibition (IACE 2015)
AIP Conf. Proc. 1691, 030015-1–030015-8; doi: 10.1063/1.4937034

expected travel time is then analyzed and compared to that of arbitrary arrival rates. As a case study, we implement our approach to a hall of a university college which is described in our previous paper [7].

This paper is organized as follows. We first introduce the mathematical background of the *M/G/C/C* approach and discuss how it measures pedestrian traffic and congestion. We then describe the structure of the hall and present the best arrival maximizing the throughput of each corridor. We next explain how the network flow programming approach can be used to find the optimal arrival rates to source corridors of a topological network. Finally, some concluding remarks are given in the last section.

M/G/C/C ANALYTICAL MODEL

An *M/G/C/C* approach considers a space (e.g., a corridor) as servers for its requesting pedestrians. The servers' service times in terms of walking speeds are determined by the current number of residing pedestrians in the space. If the space is full, pedestrians are blocked from entering the space and they thus have to queue until it is available again. The effect of the number of pedestrians to the current walking speed was formulized by Yuhaski and Smith [3]. They presented linear and exponential models of walking speed as follows:

$$\text{Linear: } V_n = \frac{V_1}{c}(c+1-n) \tag{1}$$

$$\text{Exponential: } V_n = A\exp\left[-\left(\frac{n-1}{\beta}\right)^{\gamma}\right] \tag{2}$$

$$\text{where } \gamma = \frac{\ln\left[\dfrac{\ln(V_a/V_1)}{\ln(V_b/V_1)}\right]}{\ln\left(\dfrac{a-1}{b-1}\right)},$$

$$\beta = \frac{a-1}{\left[\ln\left(\dfrac{V_1}{V_a}\right)\right]^{1/\gamma}} = \frac{b-1}{\left[\ln\left(\dfrac{V_1}{V_b}\right)\right]^{1/\gamma}},$$

γ, β = Shape and scale parameters for the exponential model,
V_n = Average walking speed for n pedestrians in a corridor,
V_a = Average walking speed when crowd density is 2 peds/m^2 = 0.64 m/s,
V_b = Average walking speed when crowd density is 4 peds/m^2 = 0.25 m/s,
V_1 = Average walking speed for a single pedestrian = 1.5 m/s,
n = Number of pedestrians in a corridor,
$a = 2 \times l \times w$,
$b = 4 \times l \times w$,
$c = 5 \times l \times w$,
l = corridor length in meters, and
w = corridor width in meters.

Based on the models, Yuhaski and Smith [3] developed the limiting probabilities for the number of pedestrians in an *M/G/C/C* model as follows:

$$P_n = \frac{[\lambda E(S)]^n}{n!f(n)f(n-1)\ldots f(2)f(1)}P_0 \quad n = 1, 2, 3, \ldots, c \tag{3}$$

where, $P_0^{-1} = 1 + \sum_{n=1}^{C} \left[\frac{[\lambda E(S)]^i}{i! f(i) f(i-1) \ldots f(2) f(1)} \right]$.

λ is the arrival rate to a corridor, *E(S)* is the expected service time of a single pedestrian in the corridor; i.e., *E(S)*= *l*/1.5, P_n is the probability when there are *n* pedestrians in the corridor, P_0 is the probability when there is no pedestrian in the corridor, and *f(n)* is the service rate and is given by $f(n) = \frac{V_n}{V_1}$. *c* meanwhile refers to the capacity of the corridor. Pedestrians attempting to enter the corridor with full capacity will be blocked. The probability of such blocking (P_{balk}) is equal to P_n where *n* equals to *c*. *M/G/C/C* networks have been shown equal to *M/M/C/C* networks by Cheah and Smith [8]. Thus, various performance measures of the corridor can then be computed as:

$$\theta = \lambda(1 - P_{balk}), \quad E(N) = \sum_{n=1}^{C} nP_n \quad and \quad E(T) = \frac{E(N)}{\theta}. \tag{4}$$

θ is the throughput of the corridor (in pedestrians per second; i.e., ped/s), *E(N)* is the expected number of pedestrians in the corridor and *E(T)* is the expected service time in seconds.

A UNIVERSITY COLLEGE HALL AS A CASE STUDY

We consider a university college hall as a case study of our *M/G/C/C* and network flow programming approaches. Its layout is shown in Figure 1. The numbers represent the corridors, the alphabets *A* to *I* represent the entrance doors to relevant corridors and *A'*, *B'* and *C'* are the exits to open spaces. Exits *A'* and *B'* are the exit corridors while exit *C'* is a staircase to the open space. Table 1 represents the dimensions of the corridors in terms of their lengths and widths in meters.

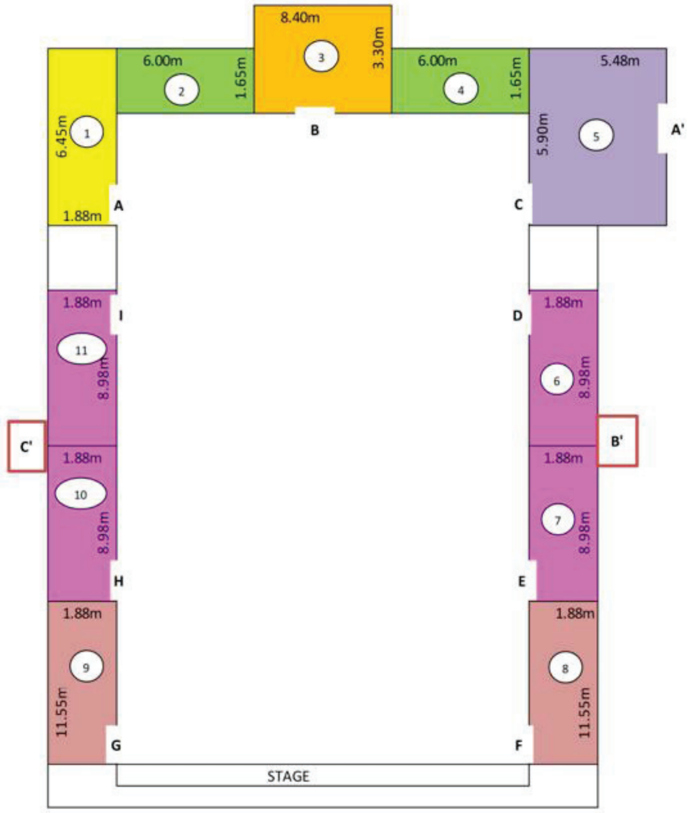

FIGURE 1. A University College's Hall Structure

There are 13 corridors in the hall. Corridors 1, 3, 5, 6, 7, 8, 9, 10 and 11 are source corridors. Corridors 2 and 4 are intermediate corridors. Occupants choose their nearest source corridors to exit the hall using doors *A*, *B*, *C*, *D*, *E*, *F*, *G*, *H* and *I*. They then travel to surrounding open spaces through exits *A'*, *B'* and *C'*. Occupants from doors *A*, *B* and *C* choose exit *A'*. Occupants from doors *D*, *E* and *F* choose exit *B'* while occupants from doors *I*, *H* and *G* choose exit *C'*. The relationship between the arrival rates (λ) and throughputs (θ) of the corridors when the hall is considered as a network is presented in Table 1. The table also shows the optimal arrival rate and its impacts to the throughputs, blocking probability (p(c)), expected number of occupants (L) and average travelling time (W) of a relevant corridor.

TABLE 1. Dimensions, Relationships, Optimal Arrival Rates and Performances of Corridor

Corridor		Length	Width	λ	Total λ	θ	Best λ	Best θ	p(c)	L	W
	1	6.45	1.88	λ_1	λ_1	θ_1	1.99718	1.95116	0.02304	19.60526	10.04801
	3	8.40	3.30	λ_3	$\lambda_3 = \lambda_3 + \theta_2$	θ_3	3.55649	3.52221	0.00964	37.26319	10.57950
	5	5.48	1.77	λ_5	$\lambda_5 = \lambda_5 + \theta_4$	θ_5	4.12374	4.07036	0.01294	29.74330	7.30729
	6	8.98	1.88	λ_6	λ_6	θ_6	2.01882	1.98558	0.01647	25.26585	12.72465
Source	7	8.98	8.98	λ_7	$\lambda_7 = \lambda_7 + \theta_8$	θ_7	2.01882	1.98558	0.01647	25.26585	12.72465
	8	11.55	1.88	λ_8	λ_8	θ_8	2.01792	1.99291	0.01240	30.33084	15.21941
	9	11.55	1.88	λ_9	λ_9	θ_9	2.01792	1.99291	0.01240	30.33084	15.21941
	10	8.98	1.88	λ_{10}	$\lambda_{10} = \lambda_{10} + \theta_9$	θ_{10}	2.01882	1.98558	0.01647	25.26585	12.72465
	11	8.98	1.88	λ_{11}	λ_{11}	θ_{11}	2.01882	1.98558	0.01647	25.26585	12.72465
	2	6.00	1.65	λ_2	$\lambda_2 = \theta_1$	θ_2	1.76335	1.71053	0.02995	17.61650	10.29886
	4	6.00	1.65	λ_4	$\lambda_{4'} = \theta_3$	θ_4	1.76335	1.71053	0.02995	17.61650	10.29886
	A'	-	-	$\lambda_{A'}$	$\lambda_{A'} = \theta_5$	$\theta_{A'}$	-	-	-	-	-
Exiting	B'	3.60	4.00	$\lambda_{B'}$	$\lambda_{B'} = \theta_6 + \theta_7$	$\theta_{B'}$	4.30450	4.21867	0.01994	22.84601	5.41545
	C'	10.00	3.00	$\lambda_{C'}$	$\lambda_{C'} = \theta_{10} + \theta_{11}$	$\theta_{C'}$	3.25133	3.22194	0.00904	40.39662	12.53799

Using the optimal λ in Table 1, we then use the network flow programming model to find the optimal arrival rates of source corridors maximizing the total throughput of the hall. Thus, the objective function of the model is to maximize the total flow out of the exit corridors subject to the flow out of each corridor is equal to its flow in and the maximum arrival rate of each corridor must less than or equal to its optimal arrival rate.

NETWORK FLOW PROGRAMMING FOR THROUGHPUT OPTIMIZATION

In the network flow programming model, each corridor is represented using a node while the links between corridors are presented using edges. The network could have multiple source nodes ($s_1, s_2, \ldots, s_n$) and sink nodes ($t_1, t_2, \ldots, t_n$). We thus introduce a fictitious super-source node S and a fictitious super-sink node T. The flow capacities from S into each available source node (S, s_i) and the flow capacities from each available sink node into T (t_j, T) are unlimited; i.e., $c(S, s_i) = c(t_j, T) = \infty$.

The network flow programming has the following characteristics:

Notation

i = an index for origin node i

j = an index for destination node j

Decision variable

$x_{i\to j}$ = the flow from origin node i to destination node j

Mathematical Formulation

Maximize $x_{T\to S}$	$x_{T\to S}$ represents the flow from super-sink node T back to super-source node S
subject to:	
$\sum_j x_{i\to j} - (\sum_i x_{j\to i} - x_{T\to S}) = 0$	outflow of node i must equal to its inflow for node $i = S$
$\sum_j x_{i\to j} - (\sum_i x_{j\to i} - x_{S\to i}) = 0$	outflow of source node i must equal to its inflow for every node $i = s_1, s_2, \ldots, s_n$
$\sum_j x_{i\to j} - \sum_i x_{j\to i} = 0$	outflow of node i must equal to its inflow for every node $i \neq s, t$
$(\sum_j x_{i\to j} + x_{i\to T}) - \sum_i x_{j\to i} = 0$	outflow of sink node i must equal to its inflow for every node $i = t_1, t_2, \ldots, t_n$
$(\sum_j x_{i\to j} + x_{T\to S}) - \sum_i x_{j\to i} = 0$	outflow of node i must equal to its inflow for node $i = T$
$x_{i\to j} \le u_{i\to j}$	flow capacity for every edge $i\to j$
$x_{i\to j} \ge l_{i\to j}$	minimum flow for every edge $i\to j$

Note that the summation of arrival rates to a corridor must be smaller or equal to its optimal arrival rate; i.e., $\sum_i x_{i\to j} \le c_j$ where c_j is the optimal arrival rate to corridor j.

The network flow programming model for the hall using the LINGO software is shown in Table 2. Note that we first write the network's objective function; i.e., to maximize the pedestrian flow from super-sink node T back to super-source node S (line 2). The network has nine source corridors. Thus, its *XT_S* is the summation of the arrival rates of source Corridors 1, 3, 5, 6, 7, 8, 9, 10 and 11(lines 5 and 6). We then ensure that all flow out of each node is equal to its flow in (lines 9 to 21). For example, the flow from Corridor 1 must equal to the flow to Corridor 2 (line 9). Third, the flow out of the super sink node *T* must equal to its flow in; i.e. the throughput of Corridors 5, *B'* and *C'* must equal must equal to *XT_S* (line 24). Fourth, we ensure that the upper limit of an arrival rate of a node must be smaller or equal to its optimal arrival rate to guarantee that no blocking exists (lines 27 to 39). For example, the arrival rate to Corridor 3; i.e., the summation of arrival rate of Corridor 3 and the throughput of Corridor 2 must be smaller or equal to its optimal arrival rate; i.e., 3.5565 ped/s (line 29).

TABLE 2 Network Flow Programming Model for the Hall

```
1  MODEL:
2        [R_OBJ] MAX = XT_S;
3
4        !Flow out of a super source node equals its flow in;
5         XT_S = XS_Corr1 + XS_Corr3 + XS_Corr5 + XS_Corr6 + XS_Corr7 +
6         XS_Corr8 + XS_Corr9 + XS_Corr10 + XS_Corr11;
7
8        !Flow out of each node equals its flow in;
9         XS_Corr1 = X_Corr1_Corr2;
10        X_Corr1_Corr2 = X_Corr2_Corr3;
11        XS_Corr3 + X_Corr2_Corr3 = X_Corr3_Corr4;
12        X_Corr3_Corr4 = X_Corr4_Corr5;
13        XS_Corr5 + X_Corr4_Corr5 = X_Corr5_T;
14        XS_Corr6 = X_Corr6_CorrBbar;
15        XS_Corr7 + X_Corr8_Corr7 = X_Corr7_CorrBbar;
16        XS_Corr8 = X_Corr8_Corr7;
17        XS_Corr9 = X_Corr9_Corr10;
18        XS_Corr10 + X_Corr9_Corr10 = X_Corr10_CorrCbar;
19        XS_Corr11 = X_Corr11_CorrCbar;
20        X_Corr6_CorrBbar + X_Corr7_CorrBbar = X_CorrBbar_T;
21        X_Corr11_CorrCbar + X_Corr10_CorrCbar = X_CorrCbar_T;
22
23       !Flow out of a super sink node equals its flow in;
24        X_Corr5_T +       X_CorrBbar_T +    X_CorrCbar_T = XT_S;
25
26       !Maximum arrival rate for a node;
27        XS_Corr1 <= 1.9972;
28        X_Corr1_Corr2 <= 1.7633;
29        XS_Corr3 + X_Corr2_Corr3 <= 3.5565;
30        X_Corr3_Corr4 <= 1.7633;
31        XS_Corr5 + X_Corr4_Corr5 <= 4.1237;
32        XS_Corr6 <= 2.0188;
33        XS_Corr7 + X_Corr8_Corr7 <= 2.0188;
34        XS_Corr8 <= 2.0179;
35        XS_Corr9 <= 2.0179;
36        XS_Corr10 + X_Corr9_Corr10 <= 2.0188;
37        XS_Corr11 <= 2.0188;
38        X_Corr6_CorrBbar + X_Corr7_CorrBbar <= 4.3045;
39        X_Corr11_CorrCbar + X_Corr10_CorrCbar <= 3.2513;
40
41  DATA:
42            @POINTER( 1) = @STATUS();
43            @POINTER( 2) = R_OBJ;
44            @POINTER( 3) = XS_Corr1;
45            @POINTER( 4) = XS_Corr3;
46            @POINTER( 5) = XS_Corr5;
47            @POINTER( 6) = XS_Corr6;
48            @POINTER( 7) = XS_Corr7;
```

```
            @POINTER( 8) = XS_Corr8;
            @POINTER( 9) = XS_Corr9;
            @POINTER( 10) = XS_Corr10;
            @POINTER( 11) = XS_Corr11;
      END DATA
END
SET GLOBAL 1
GO
QUIT

SOLUTION FOR THE MODEL
Status: Globally Optimal
Objective = 11.4126
XS_Corr1 = 0
XS_Corr3 = 0
XS_Corr5 = 4.1237
XS_Corr6 = 2.0188
XS_Corr7 = 2.0188
XS_Corr8 = 0
XS_Corr9 = 0
XS_Corr10 = 2.0188
XS_Corr11 = 1.2325
```

RESULTS AND DISCUSSION

Solving the network flow programming model in Table 2 will give the optimal arrival rates of source corridors and the performance measures of other corridors as presented in Table 3. Thus, to maximize the throughput of the hall, we should flow the occupants directly to Corridor 5 with the arrival rate of 4.1237 ped/s to exit through Corridor *A'*. Occupants near to exit *B'* should go to Corridors 6 and 7 with 2.0188 ped/s. We should block occupants from using door *F* to enter Corridor 8 to avoid competition with other occupants from Corridors 6 and 7. The same approach goes to Corridor 9. Occupants near to exit *C'* should only use Corridors 10 and 11 to exit to the open spaces with 2.0188 ped/s and 1.2325 ped/s respectively. The maximum total throughput of the hall is 11.2493 ped/s.

TABLE 3. Arrival Rates Maximizing the Total Throughput of the Hall

Corridor	Lambda	Theta	Blocking	E(N)	E(T)
1	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000
5	4.1237	4.07036	0.01294	29.74026	7.30654
6	2.0188	1.98558	0.01645	25.26272	12.72308
7	2.0188	1.98558	0.01645	25.26272	12.72308
8	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000
10	2.0188	1.98558	0.01645	25.26272	12.72308
11	1.2325	1.2325	0.00000	9.30706	7.55137
B′	3.97116	3.97015	0.00025	15.40113	3.87923
C′	3.21808	3.20884	0.00287	37.02719	11.53914
Total throughput of the network:			11.2493		

To show that 11.2493 ped/s is the maximum throughput of the hall, we compare the throughput obtained by setting arbitrary arrival rates to source corridors. For this, we set arrival rates of Corridors 1, 3 and 5 to 1.0000 ped/s, Corridors 6, 7 and 8 to 2.0000 ped/s and Corridors 9, 10 and 11 to 3.0000 ped/s. The total throughput of the network is only 8.6787 ped/s as shown in Table 4.

TABLE 4. Arbitrary Arrival Rates and Their Impacts to the Performance of the Hall

Corridor	Lambda	Theta	Blocking	E(N)	E(T)
1	1.0000	1.0000	0.0000	5.1634	5.1634
2	1.0000	1.0000	0.0000	5.0082	5.0082
3	2.0000	2.0000	0.0000	13.6785	6.8392
4	2.0000	1.4181	0.2910	44.3211	31.2542
5	2.4181	2.4181	0.0000	10.9573	4.5314
6	2.0000	1.9814	0.0093	23.0891	11.6530
7	3.9880	1.4538	0.6354	84.4171	58.0651
8	2.0000	1.9880	0.0060	27.8811	14.0249
9	3.0000	1.4434	0.5189	109.0486	75.5480
10	4.4434	1.4516	0.6733	84.5087	58.2181
11	3.0000	1.4638	0.5121	84.0132	57.3950
B′	3.4352	3.4352	0.0000	11.7914	3.4325
C′	2.9154	2.9154	0.0000	29.3894	10.0809
Total Throughput of the network:		8.7687			

CONCLUSION

This paper discusses how to find the optimal arrival rates to source corridors of a topological network maximizing its throughput. Any higher arrival rates than the optimal values will over utilize the downstream networks and cause congestions along the travelling paths rather than improving its throughput. Otherwise, any lower arrival rates than the optimal values will smoothly flow pedestrians but this will under utilize the downstream networks and eventually cause less throughput. Thus, controlling arrival rates to source corridors is important to maximize the throughput of a topological network.

ACKNOWLEDGMENT

This study was supported by the Research Acculturation Grant Scheme (RAGS) [account number 12685], Universiti Utara Malaysia. We wish to thank Universiti Utara Malaysia for the financial support. The funder had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

REFERENCE

1. F. R. B. Cruz, J. M. Smith and R. O. Medeiros, Computers and Operations Research **32**(4), 919-941 (2005).
2. A. Weiss, L. Williams and J. M. Smith, Journal of Mathematical Modelling and Algorithms **11**(4), 361-386 (2012).
3. S. J. Yuhaski Jr and J. M. Smith, Queueing Systems **4**(4), 319-338 (1989).
4. R. Khalid, M. K. M. Nawawi, L. A. Kawsar, N. A. Ghani, A. A. Kamil and A. Mustafa, PLoS ONE **8**(4), 1-9 (2013).
5. R. Khalid, M. K. Nawawi, L. Kawsar, N. Ghani, A. Kamil and A. Mustafa, Discrete Event Dynamic Systems, 1-35 (2015).
6. R. Khalid, M. A. Baten, M. K. M. Nawawi and N. Ishak, Journal of Advanced Transportation, 1-24 (2015).
7. N. Ishak, R. Khalid, M. A. Baten and M. K. M. Nawawi, AIP Conference Proceedings **1635**, 566-571 (2014).
8. J. Cheah and J. M. Smith, Queueing Systems **15**(1-4), 365-386 (1994).